

REMARKS

Claims 1-22 are pending. The claims have been amended to remove reference numerals and claims 1, 18, 21, and 22 have been amended to add commas to improve readability.

In the outstanding Office Action, the Examiner includes four rejections: (1) claims 1-22 are rejected under 35 U.S.C. §103(a) over a combination of Bayeh (U.S. Patent No. 6,098,093) and Freund et al., U.S. Patent No. 5,925,098; (2) claims 1-5, 9-11, 15, 16, 18, 21, and 22 are rejected under 35 U.S.C. §102(b) as being anticipated by Freund and claims 6-8, 12-14, 17, 19, and 20 are rejected as being obvious over Freund; (3) claims 1-3, 6, and 9 are rejected as being anticipated by Hanif et al., U.S. Patent No. 5,796,954; and (4) claims 1-22 are rejected as being anticipated by Ganmukhi, U.S. Patent No. 5,850,399. Applicants respectfully traverse these rejections.

Claims 1, 18, 21, and 22 are independent claims, and each of these independent claims recites similar subject matter. Claim 1 is representative. Claim 1 recites the following:

A method of managing a plurality of sessions, the sessions being between a plurality of terminals and a server having a plurality of threads, the method comprising:

grouping the sessions into a plurality of groups; and

assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.

It is respectfully submitted that none of the cited art includes the unique features of independent claim 1 and in particular the subject matter of “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.” It is noted that a thread is assigned to each group of *sessions*, which means that the groups correspond to multiple sessions in the subject matter of “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions”.

With regard to the rejections in (1), Bayeh is directed to spreading requests among a number of servlets. Because requests are spread among a number of servlets such that any servlet can handle requests from any session, more than one servlet might be able to access — at the same time — session information for a particular session. Bayeh discloses techniques for ensuring that only one servlet can access session information at any time for a particular session while requests can still be directed to any servlet. See, e.g., FIGS. 3, 4A, and 4B of Bayeh, and in particular steps 410-480. Furthermore, in Bayeh, the requests are passed through a load balancing host 59, which sends the requests to web servers 60, 62, and 64. The load balancing host 59 sends requests to a “server selected according to policies implemented in the load-balancing host software.” Bayeh, col. 8, lines 52-53. It is believed that the “policies” are based on load of the server and requests are sent to a server based on load. The load balancing host 59 is not disclosed as one that would “group” the requests based on session. Consequently, there is no disclosure or implication in Bayeh of “grouping the sessions into a plurality of groups” or “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions” as recited in claim 1.

Freund also does not disclose or imply the recited subject matter from independent claim 1. Instead, what Freund appears to disclose is a system for ensuring that all related requests (e.g., related through a specific transaction) are sent to the same thread. See Freund at col. 5, lines 23-27; and col. 6, lines 39-43. Freund describes a transaction as the following: “A transaction defines a single unit of work that must either be fully completed or fully purged without action”. Freund, col. 3, lines 11-12. Freund also states the following: “According to these various embodiments, a scheduling mechanism ... ensures that all requests that are related (e.g. part of the same transaction) are sent to the same execution thread for processing.” Freund, col. 6, lines 39-43. There is no statement or implication in Freund that multiple transactions are assigned to a single thread. In fact, it appears in Freund that a single thread is assigned to a single transaction: “Also, the scheduling mechanism can isolate the execution thread for a particular transaction by *not allowing requests unrelated to that transaction from being processed* on the transaction's assigned execution thread.” Freund, col. 7, lines 6-10 (emphasis added).

For at least these reasons, none of Bayeh, Freund, or their combination discloses or implies the subject matter from claim 1 of “grouping the sessions into a plurality of groups” or “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions”.

Furthermore, there does not appear to be motivation for combination of Bayeh with Freund, as Bayeh specifically allows requests from a single session to be spread amongst a number of servlets and therefore provides a mechanism for allowing multiple servlets to access session information from a single session. By contrast, the system in Freund appears to send all requests from a single transaction to a single thread. Therefore, the system of Freund would not need the mechanism of Bayeh, as only a single thread in Freund would handle requests from a single session. In Freund, multiple threads would *not* attempt to access session information from a single session, whereas allowing multiple threads to access session information for a single session is the main idea in Bayeh.

Therefore, there is no motivation to combine Bayeh and Freund and the combination of Bayeh and Freund is invalid. Because the combination is invalid, even if Bayeh and Freund disclose all elements in independent claim 1 (which Applicants submit are not disclosed by a combination of Bayeh and Freund), the rejection under §103 is improper because of an inappropriate combination of Bayeh and Freund.

With regard to the rejections in (2), as stated above, Freund does not appear to disclose the subject matter in independent claim 1 and, in fact, appears to teach the opposite, where one thread only handles requests from one transaction and a single thread is assigned to a single transaction. By contrast, claim 1 recites the subject matter of “assigning a thread to each group of *sessions* so that the assigned thread only handles the events of that group of sessions” (emphasis added). Therefore, Freund does not disclose at least the subject matter of “grouping the sessions into a plurality of groups” and “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions” in claim 1.

With regard to the rejections in (3), Hanif also does disclose the subject matter in independent claim 1. In Hanif, certain types of requests are routed to a session listening socket (SLS) and other types of requests are routed to a server session socket (SSS). See Hanif, col. 4, lines 12-56. There is one SLS to handle all sessions. Hanif, col. 4, lines 43-49; col. 5, lines 26-28. There is one SSS for each of N sessions. Hanif, col. 4, lines 50-56; col. 5, lines 26-28. One queue 190 is used to handle requests from the SLS 170. Hanif, col. 5, lines 19-59 and FIG. 6 and in particular lines 23-25 and 56-59. One queue 200 is used to handle requests from the N SSS 180. Hanif, col. 5, line 19 to col. 6, line 4 and FIG. 6 and in particular lines 23-25 of col. 5 and col. 5, line 64 to col. 6, line 4. Local threads 202 handle requests from only local queue 190, while global threads 204 handle requests from only global queue 200. Hanif, col. 5, lines 31-34.

What is apparent in Hanif is that the local threads 202 handle all requests from the SLS 170 and therefore are not assigned to certain sessions from the N sessions. See, e.g., Hanif, col. 5, lines 25-34. Furthermore, in Hanif, the global threads 204 are assigned to requests in the queue 200, which can contain requests from any of the N sessions. Col. 5, line 66 to col. 6, line 1; col. 7, lines 8-11. Therefore, the global threads 204 are not assigned to groups of sessions and instead can handle requests from any session. For at least these reasons, Hanif does not disclose at least the subject matter of “grouping the sessions into a plurality of groups” and “*assigning a thread to each group of sessions* so that the assigned thread only handles the events of that group of sessions” in claim 1.

Regarding the rejections in (4) above, Ganmukhi appears inapplicable to the subject matter in independent claim 1. However, Ganmukhi does use the term “session”, which relates to voice, video, or data transmissions (see, e.g., Ganmukhi at col. 2, lines 11-12). The Examiner appears to equate a “thread” of claim 1 with a “scheduler” (i.e., 20-70; see FIG. 1 of Ganmukhi) of Ganmukhi. Even if this equating is true (which it does not appear to be), claim 1 specifically states that the sessions are between a number of terminals and a server having threads. In Ganmukhi, the transmissions appear to be between a source not shown or described and a destination also not shown or described. For instance,

Ganmukhi states that “[t]he presently disclosed hierarchical scheduler is particularly useful in an ATM network since the hierarchical scheduler adapts the user traffic to the cell based network.” Col. 3, lines 19-22. The “scheduler 10” of Ganmukhi appears to be an intermediate device between some source having user traffic and an ATM network. While it is unclear whether the user traffic is from a number of terminals, there is no disclosure in Ganmukhi of “a plurality of sessions, the sessions being between a plurality of terminals and *a server having a plurality of threads*” as there are no “sessions” between a plurality of terminals and the hierarchical scheduler of FIG. 1 of Ganmukhi because the user traffic is meant for some destination other than the hierarchical scheduler of Ganmukhi.

Furthermore, Applicants define a session as the following: “A session is a series of interactions between a terminal and a server having a well-defined beginning and end and involving agreed-upon characteristics.” Any such “session” in Ganmukhi is between sources and destinations not shown or described in Ganmukhi.

Therefore, Ganmukhi does not disclose at least “managing a plurality of sessions, the sessions being between a plurality of terminals and a server having a plurality of threads”, “grouping the sessions into a plurality of groups” and “assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions” in claim 1. Claim 1 is patentable over Ganmukhi.

Moreover, the Examiner asserts that “Claims 2-21 are either expressly taught by Ganmukhi or are inherent variations thereof”. For purposes of future appeals, Applicants respectfully request that more information be given with respect to this rejection. For instance, Applicants cannot find the term “Internet” in Ganmukhi and Ganmukhi appears unrelated to conducting transactions on the Internet, which means that the subject matter of claim 16, reciting “A method according to claim 1 in which the sessions involve obtaining information or conducting transactions through the Internet”, is not “expressly taught” or “inherent” to Ganmukhi.

Claim 1 is therefore patentable over Bayeh and Freund in (inappropriate) combination, Freund, Hanif, and Ganmukhi. Because claim 1 is patentable over these cited references, its dependent claims 2-17 are also patentable for at least the reasons give with respect to claim 1. Therefore, claims 2-17 are patentable over Bayeh and Freund in (inappropriate) combination, Freund, and Ganmukhi, and claims 2, 3, 6, and 9 are patentable over Hanif.

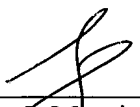
Each independent claim 18, 19, and 20 contains similar subject matter to independent claim 1. For instance, claim 18 recites “A server for managing a plurality of sessions with a plurality of terminals, the server comprising a plurality of threads, grouping means to group the sessions into a plurality of groups, and assigning means to assign a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.” Claim 21 recites “A communications system comprising a server and a plurality of terminals the server and the terminals conducting a plurality of sessions the server comprising a plurality of threads, grouping means to group the sessions into a plurality of groups and assigning means to assign at least one thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.” Claim 22 recites “A computer program product for managing a plurality of sessions the sessions being between a plurality of terminals and a server having a plurality of threads, comprising: computer readable program means for grouping the sessions into a plurality of groups; and computer readable program means for assigning a thread to each group of sessions so that the assigned thread only handles the events of that group of sessions.”

The arguments given above with respect to independent claim 1 are equally valid with respect to independent claims 18, 21, and 22. Therefore, Applicants respectfully submit that independent claims 18, 21, and 22 are patentable over Bayeh and Freund in (inappropriate) combination, Freund, and Ganmukhi. Because claim 18 is patentable, its dependent claims 19 and 20 are also patentable over Bayeh and Freund in (inappropriate) combination, Freund, and Ganmukhi.

S.N. 10/019,330
Art Unit: 2143

Based on the foregoing arguments, it should be apparent that claims 1-22 are thus allowable over the reference(s) cited by the Examiner, and the Examiner is respectfully requested to reconsider and remove the rejections. The Examiner is invited to call the undersigned attorney for any remaining issues.

Respectfully submitted:



Robert J. Mauri
Reg. No.: 41,180

7/05/06
Date

Customer No.: 29683

HARRINGTON & SMITH, LLP
4 Research Drive
Shelton, CT 06484-6212

Telephone: (203)925-9400
Facsimile: (203)944-0245
email: rmauri@hspatent.com

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. BOX 1450, Alexandria, VA 22313-1450.

7/5/2006

Name of Person Making Deposit

Elaine F. Mauri

Date